VISIT…

LANZAROTE Caliente.COM

Нейрокомпьютеры: разработка, применение. 2003. №12. - С.???

Уточнение решения обратной задачи для нейросети-классификатора

Царегородцев В.Г.
www.NeuroPro.ru tsar@neuropro.ru

Предложены приемы коррекции и уточнения решения обратной задачи, полученного градиентным обучением входных сигналов нейросети-классификатора. В результате возможно достижение робастности, получение дополнительной информации и определение степени доверия к выданному нейросетью начальному решению.

A few simple tricks to post-correction of inverse problem solution obtained by trained neural classifier inversion are proposed. This gives more robust results, additional information about possible solutions and estimation of degree of believe of initial uncorrected decision.

Обратная задача нейромоделирования и способы решения

Рассматриваются вопросы уточнения и коррекции решения обратной задачи нейромоделирования при постановке прямой задачи как задачи классификации с учителем. В качестве нейромодели взята нейросеть, обучаемая алгоритмом обратного распространения. На целевую функцию, оптимизируемую при обучении, каких-либо ограничений не накладывается. Под обратной задачей в ее нейросетевой постановке понимаем нахождение таких значений сигналов на входе обученной сети, чтобы на выходе сети было состояние, соответствующее нужному классу. Очевидно, что даже при наличии ограничений на диапазоны изменения всех или отдельных переменных может существовать континуум решений. Поэтому используется коррекция некоторых стартовых значений входных сигналов сети алгоритмом, подобным алгоритму обучении нейросети, до момента получения требуемого состояния на выходе сети. В [1-3] предложено алгоритмом обратного распространения вычислять частные производные целевой функции (определяющей степень отклонения выхода сети от требуемого состояния) по значениям входных сигналов и вести коррекцию последних вдоль направления антиградиента. В [4] эта идея развита для рекуррентных сетей и сетей с временными задержками. Градиентный спуск приведет к одному из возможных решений обратной задачи – аттрактору для стартового вектора входных сигналов, а остановка в области локального для градиентного спуска минимума скажет, что решение обратной задачи при заданных начальных условиях или ограничениях на значения сигналов может и не существовать.

Для получения набора возможных решений известен эволюционный алгоритм [5] поиска точек, лежащих вдоль заданной линии уровня целевой или иной функции, построенной над значениями выходных сигналов сети: т.е. линией уровня может быть разделяющая классы поверхность или многообразие, на котором сигнал нейрона нужного класса на заданное значение превышает сигнал нейрона другого класса. Этот алгоритм также уравнивает плотность размещения точек вдоль искомой линии уровня (т.е. ликвидирует избыточность популяции точек еще одним способом наряду с простым отбрасыванием наименее адекватных точек) и может стартовать как с некоторого числа тем или иным образом распределенных по области значений независимых переменных точек, так и с единственной заданной стартовой.

Описания указанных идей и методов есть и в обзорной работе [6]. Здесь же дополнительно используются методы предобработки и анализа данных. Основными предлагаемыми результатами являются следующие:

1. Повышение робастности решения обратной задачи.
2. Получение набора интервальных значений параметров, описывающих границу

между классами.
3. Получение одного или нескольких «типичных» для выборки вариантов решения обратной задачи.
4. Возможность коррекции классификационной модели для обеспечения робастности решения прямой задачи и определения степени доверия к результатам решения обратной задачи в конкретной ситуации.

Есть и другие способы решения обратной задачи, например, путем [7] обучения отдельной нейросети, по значениям зависимых в терминах проблемной области признаков прогнозирующей значения независимых, или построения сети-автоассоциатора, в среднем слое которой дополнительно требуется выдача решения прямой задачи (после «разрезания» автоассоциатора по выходам узкого слоя получаем две сети, решающие прямую и обратную задачи соответственно). Здесь нет возможных при градиентном обучении [1-4] входных сигналов сети локальных минимумов и точек перегиба целевой функции, но автоассоциатор при числе нейронов в среднем слое, равному числу классов или зависимых переменных, зачастую можно и не обучить до нужной точности. Наибольшие же возражения вызывает невозможность получения одинакового интер- и экстраполяционного поведения двух нейросетей – на тех значениях независимых переменных, которые решающая обратную задачу нейросеть выдаст для заданного класса, решающая прямую задачу нейросеть может выдать другой номер класса. Поэтому этот подход далее не рассматривается.

Предварительная коррекция обучающей выборки

Необходимым предварительным шагом является кластерный анализ примеров разных классов по-отдельности. Кластеризация в пространстве независимых признаков ведется методом динамических ядер с использованием евклидовой метрики, метрики Махаланобиса внутри кластера (если число примеров каждого кластера достаточно для достоверной оценки его ковариационной матрицы) либо иной метрики, например, робастной к выбросам в данных. Далее в соответствии с [8] примеры, статистически достоверно выходящие за типичный радиус разброса примеров своего кластера, проецируются по направлению к центру кластера до момента удовлетворения этому критерию. Обучение нейросети ведется на откорректированной выборке; информация о найденных кластерах будет востребована только при решении обратной задачи.

Такая предварительная коррекция желательна и даже обязательна, поскольку:

- ликвидируются выбросы в данных, способные привести к переобучению сети;
- для быстрых алгоритмов обучения, вроде метода сопряженных градиентов, может снизиться риск переобучения в одной области значений входных признаков с одновременным недообучением в другой, или риск возникновения разных степеней переобучения в разных областях, как указано в [9];
- может снизиться чувствительность решения прямой и обратной задач к малым колебаниям значений независимых переменных.

Также можно использовать гибридный метод [10] соединения идеи динамических ядер и анализа главных компонент: с каждым ядром связывается свое пространство линейных главных компонент, описывающее относящиеся к ядру-кластеризатору точки данных, а оптимизация положений ядер и перерасчет главных компонент идут так, чтобы минимизировать в итоге дисперсию длин проекций точек на ГК. Коррекции затем подвергаются примеры, далеко отстоящие от их проекций, а проецирование идет не по направлению к ядру, а на пространство главных компонент.

При слабовыраженных или отсутствующих кластерах, отсутствии выбросов в данных или желании в дальнейшем иметь более детальное квантование данных, можно в дальнейшем вместо результатов кластерного анализа использовать дополнительно

сделанное квантование данных растущим нейронным газом [11], по своему способу построения и адаптации гарантирующим соответствие плотности распределения квантующих ядер плотности распределения точек данных. Естественно, квантование ведется по прежнему раздельно для примеров разных классов.

Обучение нейросети с применением робастной целевой функции, т.е. без предварительной коррекции выборки, тоже способно обеспечить достижение указанных свойств, но результаты кластеризации будут активно использованы в дальнейшем. Тем более, что робастные целевые функции (см., например, [12]) разработаны для задачи регрессии, а не классификации.

Уточнение нейросетевого решения обратной задачи

Решение обратной задачи, найденное методом [1-4], т.е. полученное обучением входных сигналов нейросети, заменяется на ближайший к нему вектор – ядро кластера искомого класса. Такая замена более адекватна внутренней структуре данных: результат совпадает с локальным максимумом плотности распределения точек-примеров, соответствующих кластеру. Если же новое значение вектора входных сигналов дает меньший уровень превышения сигнала нейрона-победителя над сигналами нейронов других классов, то это указывает на избыточную жесткость требований к свойствам отклика нейросети на этом примере.

Пограничные интервальное и точечные решения обратной задачи

Кластеру, центр которого взят в качестве уточненного решения обратной задачи, соответствует набор интервалов значений переменных, определенный по точкам этого кластера. Но напрямую этот набор интервалов использовать зачастую нельзя – в случае перекрывающихся распределений классов и высокой цены ошибки классификации примеров другого класса построенная нейросетью разделяющая классы поверхность может отсекать достаточно большое число точек найденного кластера.

Эволюционный алгоритм [5,6], стартующий с точки, найденной обучением входных сигналов сети, для поиска точек, лежащих вдоль соответствующей линии уровня целевой функции, позволит найти две точки, наиболее удаленные от заданной и притягивающиеся к тому же кластеру примеров искомого класса, такие, что движение далее по линии уровня будет притягиваться к другим кластерам класса.

Две эти терминальные точки зададут интервалы значений независимых переменных, в которые должно попасть состояние управляемой системы перед тем, как перейти в нужный класс. Из набора же точек, лежащих на линии уровня между терминальными, можно дополнительно выделить точку, ближайшую к точке, взятой в качестве начального состояния системы (начальным входным сигналам перед их «обучением» на сети), и точку, наиболее удаленную от нее. Эти две точки задают координаты для наиболее оптимистического и наиболее пессимистического вариантов возможных траекторий движения системы при целенаправленном воздействии на нее.

Получение нескольких вариантов решения обратной задачи

Описанные в двух предыдущих параграфах приемы можно использовать для нахождения нескольких кластеров точек искомого класса, служащих решениями обратной задачи. Для этого нужно запустить эволюционный алгоритм с точки, полученной градиентной коррекцией входов сети, и расширять интервал расположения точек на линии уровня целевой функции до тех пор, пока точки не станут притягиваться к заданному числу кластеров искомого класса. Далее можно определять интервалы возможных значений независимых переменных для кластеров, интервалы между точками на линии уровня, и объединения этих интервалов.

Простое же взятие нескольких кластеров примеров искомого класса, ближайших к первому найденному, некорректно, поскольку может дать кластеры, не лежащие вдоль разделяющей поверхности, а окруженные кластерами того же класса.

Дать более точные рекомендации здесь не представляется возможным: пользователь должен сам определять приемлемость стратегий исходя из содержательной постановки задачи, экспертных знаний о предметной области, свойств обрабатываемых выборок данных.

Коррекция классификационной модели

Перейдем от решения обратной задачи для одного примера к статистике решений обратной задачи на примерах некоторой выборки, не обязательно порожденной из той же генеральной совокупности, что и обучающая.

Пусть найдено большое число различных возможных неоткорректированных решений обратной задачи: например, для каждого класса можно найти решение обратной задачи при начальных инициализациях независимых переменных нулевыми, средними (если средние значения предобработанных переменных не равны нулю), минимальными и максимальными значениями в разных комбинациях. Дополнительно отберем из этих решений только те, на которых на выходе сети действительно получен требуемый класс. Найденные вектора решений обратной задачи анализируются.

Как указано [3,4,13], в качестве решения обратной задачи может быть найдена точка, «совсем не похожая» на использованные при обучении. Если не рассматривать случай изначальной малости обучающей выборки, то указанная ситуация соответствует решению обратной задачи, выходящему за пределы «облака» обучающих точек в пространстве независимых переменных. Если возможны содержательная, статистическая или визуальная (подобно описанной в [3,4,13] при распознавании образов) инспекции нетипичных вариантов решения, то для каждого из них можно определить, что либо это корректный вариант и нейросеть имеет навык экстраполяции решения обратной задачи, либо это некорректный образ, который нужно в дальнейшем классифицировать отдельно или не допускать его появления в качестве решения.

Активная коррекция классификационной модели была использована, например, [14] для сети Хопфилда путем добавления ложных для сети аттракторов в обучающую выборку и изучения изменения числа аттракторов и объемов их областей притяжения в новой сети. Здесь же, при необходимости гарантировать в дальнейшем правильное решение прямой и обратной задач на найденных нетипичных, но содержательно корректных образах, такие примеры с указанием соответствующих номеров классов можно добавлять в обучающую выборку и заново обучать нейросеть.

При нахождении же некорректных векторов значений независимых признаков можно изменить классификационную модель. Такие вектора включаются в обучающую выборку как примеры нового, «некорректного» класса. Кластеризация примеров этого класса не обязательна, к тому же получение достаточного для кластерного анализа числа таких примеров может быть трудоемким.

При добавлении в выборку как корректных, так и некорректных примеров возможно итерирование – после обучения нейросети можно возвращаться к поиску вариантов решения обратной задачи и добавлению примеров в выборку.

Когда обученная нейросеть будет решать прямую задачу на новых данных, то «некорректный» класс в качестве ответа сети укажет, что в качестве вектора независимых признаков почему-то поданы некорректные с содержательной для задачи точки зрения значения, и сам пользователь-эксперт ранее потребовал, чтобы в данной области значений решение классифицировалось как некорректное.

А если решение обратной задачи приводит к вектору входных сигналов,

соответствующему «некорректному» классу на выходе сети вместо искомого, то это индицирует явную низкую достоверность предлагаемого решения – скорее всего, полученный вектор значений независимых признаков будет достаточно близок именно к ранее найденным и использованным при обучении «некорректным» примерам.

Таким образом, поиском и вводом в рассмотрение «некорректных» примеров повышается степень доверия к результатам решения прямой и обратной задач. Происходит это за счет замены ожиданий хорошей или правдоподобной экстраполяции навыка решения на требование явного иного отклика сети в областях значений независимых переменных, не покрытых первоначальной обучающей выборкой.

Литература

1. *Williams R.J.* Inverting a connectionist network mapping by backpropagation of error / Proc. 8th Annual Conf. Cognitive Science Society, Hillsdale, NJ, USA, 1986. – pp.859-865.
2. *Linden A., Kindermann J.* Inversion of multilayer nets / Proc. IJCNN'1989, Washington, DC, USA, 1989. – pp.425-430.
3. *Kindermann J., Linden A.* Inversion of neural networks by gradient descent / Parallel Computing, 1990, №14. – pp.270-286.
4. *Thrun S., Linden A.* Inversion in time / Proc. EURASIP Workshop on Neural Networks, Sesimbra, Portugal, 1990.
5. *Reed R.D., Marks R.J.* An evolutionary algorithm for function inversion and boundary marking / Proc. IEEE Int. Conf. Evolutionary Computation’1995, Perth, Australia, 1995. – pp.794-797.
6. *Jensen C.A., Reed R.D., Marks R.J., El-Sharkawi M.A. et al.* Inversion of feedforward neural networks: Algorithms and applications / Proc. IEEE, 1999, Vol. 87, №9. – pp1536-1549.
7. *Yamaguchi S., Tanaka M., Itakura H.* An inverse modeling using a five-layer perceptron / Proc. IJCNN’1993, Nagoya, Japan. Vol. 3. – pp.2803-2806.
8. *Hämäläinen J.J., Järvimäki I.* Input projection method for safe use of neural networks based on process data / Proc. IJCNN'1998, Anchorage, Alaska, USA, 1998. – pp.193-198.
9. *Lawrence S., Giles C.L.* Overfitting and neural networks: Conjugate gradient and backpropagation / Proc. IJCNN’2000, Como, Italy, 2000. – pp.114-119.
10. *Kambhatla N., Leen T.* Fast nonlinear dimension reduction / Advances in Neural Information Processing Systems 6 (1993). Morgan Kaufmann, 1994. – pp.152-157.
11. *Fritzke B.* Growing self-organining networks – why? / Proc. European Symposium Artificial Neural Networks (ESANN'1996), Bruges, Belgium, 1996. – pp.61-72.
12. *Царегородцев В.Г.* Робастная целевая функция с люфтом для задач аппроксимации / Материалы XI Всеросс. конф. "Нейроинформатика и ее приложения", Красноярск, 2003. - 215с. – с.162-165.
13. *Jacobsson H., Olsson B.* An evolutionary algorithm for inversion of ANNs / Proc. Fifth Joint Conf. Information Sciences’2000. Association for Intelligent Machinery, 2000. – pp.1070-1073.
14. *Ezhov A.A.* Empty classes, predictive and clustering thinking networks / Neural Network World, 1994, №6. – pp.671-688.